FRECONWIN: UNA EXPLICACIÓN INFORMÁTICA PARA EL ESTUDIO LEXICOMÉTRICO DE TEXTOS

JOSÉ LUIS GUERRERO MARTÍN
Universitat de les Illes Balears

1. RESUMEN

El proyecto nace de una iniciativa de los departamentos del Área de Filología Francesa y el departamento de Matemáticas e Informática de la UIB, consistente en el estudio y elaboración de una herramienta informática que facilite el tratamiento de documentos de texto y permita, mediante el uso de la estadística, un análisis experimental y objetivo de la lengua para poder descubrir las estructuras de pensamiento de los autores de las obras literarias y documentos en general, facilitando su estudio y comprensión. En consecuencia, los objetivos del presente artículo son:

(a) Examinar qué datos y cálculos estadísticos pueden realmente interesarnos para llevar a cabo nuestros estudios filológicos de textos.

(b) Revisar diferentes aplicaciones existentes en el mercado relacionadas con el tema en cuestión, para obtener las características positivas de éstas.

(c) Elaborar una aplicación informática que cumpla con todos los requisitos solicitados por el departamento del Área de Filología Francesa y supere las deficiencias que se han encontrado en los programas empleados hasta este momento.

(d) Detectar los métodos estadísticos necesarios para la obtención de los resultados deseados.

(e) Implementar e integrar dichos métodos en la aplicación informática a realizar.

2. FUNDAMENTOS

Hoy en día son muchos los investigadores que han integrado el uso del análisis estadístico en la investigación cualitativa de textos literarios y eso ha quedado reflejado en un incremento de estudios que incluyen este tipo de trabajos. Recurrir a las técnicas estadísticas supone dar mayor rigor, exhaustividad y objetividad a los datos obtenidos por otras vías más intuitivas o subjetivas.

Los primeros estudios preocupados por la frecuencia con que las palabras aparecen en un texto se remontan a la antigüedad. En Alejandría, los gramáticos llegaron a elaborar listados de los *hapax*[1] de Homero, y también se conocen estudios posteriores, dirigidos a inventariar todas las palabras de la Biblia. Desde el primer tercio de este siglo, en los países anglosajones, se han analizado las concordancias (contextos discursivos en que aparece una palabra) de determinadas palabras en los grandes autores de la literatura.

En los últimos años la aplicación de la informática para realizar estudios estadísticos ha hecho que estos se desarrollen y simplifiquen de forma vertiginosa, pudiéndose almacenar, ordenar y clasificar los datos lingüísticos de manera electrónica.

La unión de la filología y la informática no es nueva, pero el desarrollo alcanzado en la última década ha sido espectacular.

El departamento del Área de Filología Francesa, dirigido por el doctor Gabriel Jordà Lliteras, desde ya hace unos años ha confiado en esta unión informática —tratamiento de textos— estadística, y con la ayuda de diversas herramientas ha ido desarrollando un método de trabajo que ha mostrado sus frutos obteniendo resultados en estudios tanto de lengua como de literatura: colección de textos, memorias y proyectos de investigación o tesis doctorales defendidas con notable éxito en distintas universidades del Estado Español.

De forma cronológica podemos decir que la forma en que el tratamiento de textos ha sido llevado a cabo en dicho departamento es el siguiente:

De una forma más o menos intuitiva, y desde luego bastante manual, el filólogo era el encargado de efectuar diversas lecturas de un documento y observar qué partes, formas, segmentos y elementos significativos del mismo eran los más importantes y en cuales había que centrarse. Se trataba de un proceso muy lento y laborioso, y que exigía mucho tiempo para extraer conclusiones que podían ser en muchos casos demasiado subjetivas. A este método podemos denominarlo como el tradicional.

La ayuda de la informática para el análisis cuantitativo de textos ha sido fundamental; sobre todo desde el momento en el cual se pudo recurrir a procesadores de texto cada vez más potentes (en primer lugar con *MS-DOS*, y posteriormente con *Windows* y/o ordenadores *Macintosh*). Con la ayuda de un scanner, y su correspondiente *OCR* (*Optical Caracter Recognizer*), se podían pasar documentos, artículos y cualquier material textual al ordenador, y a partir de ahí poder buscar ciertas palabras (formas), e ir obteniendo conclusiones, ahora con la ventaja que nos ofrece la rapidez de un ordenador en la búsqueda de textos. Con dicha información se podían crear los ficheros de resultados deseados.

Diversas empresas de software desarrollaron programas enfocados a la extracción de información sobre documentos de textos, fundamentados en la observación de las frecuen-

---

(1) Hapax es la denominación que reciben las palabras que sólo aparecen una vez en un texto o conjunto de textos considerados.

cias de las diferentes formas de un documento, y ayudados de cálculos estadísticos. Programas de este estilo son *SPADT*, *TACT*, *Hyperbase*, *Sphynx Lexica*... Estos programas, nos liberan del trabajo engorroso que suponía la realización manual de recuentos. Además con ellos podemos hacer cálculos estadísticos, y combinando varios de ellos, sacar conclusiones bastante importantes.

Entre los años 1998 y 1999, el autor del presente artículo, desarrolló una aplicación denominada *FRECON* (Frecuencias y contextos), que permitía, con una aplicación realizada en lenguaje *CLIPPER* (para ser utilizada en *PC*s con el sistema operativo *DOS*), trabajar con documentos de textos, permitiendo elaborar una relación de las diferentes palabras de un documento con sus frecuencias asociadas, buscar los contextos en los cuales aparecía una determinada palabra o forma, redirigiendo los posibles resultados a un fichero, marcar diferentes palabras del documento y redirigir los resultados a archivos... El objetivo era intentar aprovechar la información resultante para realizar con ella un documento en el cual apareciera el estudio deseado, completado con los estudios estadísticos. Para realizar dichos estudios estadísticos se recurría a aplicar el *SPADT*, *TACT* o aplicaciones de cálculo realizadas para dichos efectos también en lenguaje *CLIPPER*. Se realizaron asimismo, diversas aplicaciones reducidas de este programa original, las cuales permitían trabajar con un documento en concreto.

Los resultados obtenidos por estas aplicaciones por separado fueron bastante satisfactorios. Así pues se decidió realizar una aplicación de mayor alcance que recogiera las diferentes ventajas de las aplicaciones mencionadas anteriormente, y se integrase bajo una única aplicación. Esta aplicación pasará a llamarse de ahora en adelante, *FRECONWIN*.

3. CARACTERÍSTICAS DE FRECONWIN

Como se ha comentado anteriormente, estamos ante el diseño de una aplicación que pretende integrar un gran número de las fases del tratamiento de un documento, según una determinada metodología, con el objetivo de extraer la máxima información del mismo.

De forma muy general, consiste en una base de datos de documentos, sobre los cuales se puede hacer todo tipo de búsquedas, recuento y estudio sobre formas, segmentos y oraciones/frases de un documento (grupos sintagmáticos).

La aplicación de métodos estadísticos nos ayudará a decidir qué formas y segmentos son las más representativas (tanto por el hecho de ser infrautilizadas, como por ser sobreutilizadas), así como los que presenten un cierto grado de homogeneidad en todo el documento, y crear categorías de formas con las que se pueden llevar a cabo estudios independientes.

Los estudios podrán realizarse tanto sobre un fichero en cuestión, como sobre la combinación de diversos ficheros. Si se trata de trabajar sobre un único fichero, dispondremos de dos posibilidades:

1) Que se realice un proceso previo de identificar diferentes partes del documento.
2) Que el programa divida automáticamente el fichero en tantas partes como indique el usuario.

La primera opción parece la más interesante, ya que es el usuario el que se encarga de dividir las partes, de forma que se identifican partes realmente significativas.

3.1. Cálculos estadísticos.

La unión de la Informática con la Estadística nos proporciona un amplio abanico de diferentes tipos de análisis y resultados. Todos ellos aportan, en mayor o menor grado, datos de interés para el filólogo. Ahora bien, en la labor de concretar nuestro campo de trabajo, tanto en lengua como en literatura, nosotros nos centramos sólo en el análisis de textos y nos interesamos únicamente en los *modelos (patterns)* definidos por un uso homogéneo o por una sobre/infrautilización significativos. Éste es pues el objetivo de la herramienta informática: conseguir las máximas prestaciones para poder llevar a cabo los análisis de textos a partir de los resultados antes mencionados.

Así pues, los cálculos realizados por la aplicación se pueden clasificar básicamente en dos tipos: obtención de elementos homogéneos y determinación de especificidad. Mediante el primer método podemos obtener qué elementos de uno o varios documentos se distribuyen de modo más homogéneo entre ellos o entre todas las partes del mismo y mediante la determinación de especificidad podemos averiguar cuáles son los elementos de cada una de las partes del documento sobreutilizadas e infrautilizadas. Por elemento podemos entender cualquier parte del documento del cual estamos interesados obtener información: formas, lemas, segmentos, categorías, grupos sintagmáticos...

Hemos decidido recurrir a los métodos estadísticos del contraste de homogeneidad y determinación del cálculo de las especificidades siguiendo la ley de la distribución hipergeométrica, para solventar nuestros problemas, opción apoyada y validada por el doctor Ángel Igelmo Ganzó, profesor catedrático de la Universitat de les Illes Balears.

3.1.1. Determinación de elementos homogéneos.

Se trata de comparar las apariciones de un determinado elemento, respecto a cada una de las partes del documento, y determinar si se distribuyen de una forma más o menos homogénea, y aceptable dentro del umbral que determinemos.

Para la determinación de elementos homogéneos, recurrimos al método estadístico del contraste de homogeneidad, en concreto el χ^2 (*Chi-cuadrado*).

3.1.2. Determinación de especificidad (formas características).

El método de las especificidades consiste en determinar qué elementos son característicos de cada una de las partes de uno o varios documentos (elementos sobreutilizados —elementos de reiterada aparición—, e infrautilizados —elementos que destacan por su rareza—).

Para ello, lo que hacemos es realizar un estudio comparativo de cada una de estas partes respecto al total del documento. Básicamente consiste en efectuar una serie de cálculos matemáticos partiendo de las frecuencias reales de aparición y comparando respecto al valor que sería esperado obtener, si fijado un determinado umbral, siguiera la distribución hipergeométrica. Según el valor obtenido, podemos pensar que se trata de formas cuya desviación respecto al valor esperado no se deben al azar, y son por lo tanto características del documento.

Las formas que presentan una especificidad positiva dentro de una parte del corpus son las que se emplean por encima de lo que cabría esperar si las apariciones de ésta se distribuyeran de forma aleatoria en todo el corpus. Por el contrario, las especificidades negativas

corresponden a las formas que están infrautilizadas en relación a su presencia en todo el corpus.

Lafon (1980) desarrolló el cálculo de las especificidades siguiendo la ley de la distribución hipergeométrica y demostrando que ésta se adapta a la perfección al campo actual de la lexicometría. Éste es pues el método utilizado en el *FRECONWIN*.

Pretender atacar el problema mediante el cálculo directo de la hipergeométrica (mediante números combinatorios), no es el método más adecuado para un ordenador, ya que los valores resultantes intermedios son demasiado elevados. El método que seguimos aplica técnicas de logaritmos para intentar disminuir el valor de los números tratados.

3.2. Aplicación de los cálculos estadísticos.

Basándose en esta idea de la homogeneidad y especificidad, la aplicación informática ofrecerá los tipos de estadística que se muestran a continuación. Cuando hablamos de número de ficheros, entendemos archivos de texto independientes.

Aún así, tiene sentido el hecho de comparar dos o más documentos diferentes, ya que estos documentos pueden tener una relación entre sí; véase capítulos de un libro, artículos de un determinado autor, estudios sobre una misma materia,... Únicamente exigimos que si dichos documentos han sido sometidos a una previa desambiguación, en todos ellos se haya procedido de una forma similar y coherente.

Nº DE DOCUMENTOS	TIPO DE ESTADÍSTICA	INFORMACIÓN OBTENIDA
1	**Estadística según un determinado historial de formas**, es decir, todas las formas léxicas correspondientes con un determinado lema (ej: am%, %ar, o incluso la relación de todas las palabras) Obs: am%, es una forma de expresar todas las formas léxicas que comienzan por am, y %ar es una forma de expresar todas las formas léxicas que acaban por %ar. Básicamente podemos decir que % actúa como carácter comodín. Empleando los caracteres comodines % y _ , podemos obtener infinidad de combinaciones.	1) Lista de formas léxicas homogéneas 2) Relación de formas específicas de cada parte del documento (formas léxicas sobreutilizadas y formas léxicas infrautilizadas)
1	**Estadística según un determinado historial de segmentos**, es decir, todas la combinación de formas léxicas correspondientes con una combinación de forma polo y forma segmento (ej: mar% / sol%)	1) Lista de segmentos homogéneos 2) Relación de segmentos específicos de cada parte.

1	**Estadística de formas léxicas correspondientes a una determinada categoría.** Formarán parte de una categoría, todas aquellas formas léxicas que compartan características comunes, que son las que precisamente identifican a la categoría.	1) Lista de formas léxicas homogéneas correspondientes a una determinada categoría 2) Relación de formas léxicas específicas de cada parte (formas léxicas sobreutilizadas y formas léxicas infrautilizadas)
1	**Estadística de una determinada categoría de un documento**	1) Nos informa sobre si el uso de una categoría concreta es homogénea respecto a todas las partes del documento.
1	**Grupos sintagmáticos de 'n' formas léxicas consecutivas**	1) Relación de grupos sintagmáticos homogéneos. 2) Relación de grupos sintagmáticos específicos de cada una de las partes del documento (grupos sintagmáticos sobreutilizados e infrautilizados)
1	**Combinaciones de segmentos separados hasta una distancia determinada.**	1) Relación de segmentos homogéneos 2) Relación de segmentos específicos para cada una de las partes del documento (segmentos sobreutilizados e infrautilizados)
n	**Estadística según un determinado historial de formas**, es decir, todas las formas léxicas correspondientes con un determinado lema (ej: am%, %ar, o incluso la relación de todas las palabras), comparando entre los diversos documentos.	1) Lista de formas léxicas homogéneas 2) Relación de formas específicas de documento correspondientes al lema solicitado (formas léxicas sobreutilizadas y formas léxicas infrautilizadas)
n	**Estadística de las formas léxicas correspondientes a una determinada categoría**, comparando entre los diversos documentos.	1) Lista de formas léxicas homogéneas 2) Relación de formas específicas de cada documento correspondientes a la categoría solicitada (formas léxicas sobreutilizadas y formas léxicas infrautilizadas)

Muy posiblemente algunas de las estadísticas mencionadas anteriormente tendrán un valor realmente poco significativo, y en algunos casos de imposible aplicación, como es el caso de homogeneidad referente a formas que aparezcan un número muy bajo de veces, pero son estadísticas que consideramos útiles disponer de ellas para los casos en que ofrezcan información útil.

4. DESARROLLO DEL PROYECTO Y PRESTACIONES

Detallamos a continuación las diferentes fases del desarrollo de *FRECONWIN* y la problemática surgida en su elaboración.

En todo momento se ha pensado realizar dicha aplicación para entorno Windows, y el lenguaje de desarrollo ha sido Delphi, ya que se trata de un lenguaje de programación bastante potente y de muy amplia difusión, y que ofrece una velocidad muy razonable para la ejecución de las aplicaciones.

4.1. Elección de un sistema gestor de bases de datos.

Uno de los problemas mayores fue el decidir qué sistema gestor de base de datos emplear para almacenar toda la información que se necesita de forma permanente.

Antes de profundizar sobre las diferentes opciones sobre las que se planteó su desarrollo, intentaremos explicar algunos conceptos referentes a bases de datos, que en muchas ocasiones se confunden, y que es importante que el lector sepa diferenciar:

—Base de datos: Es una colección de datos organizados. De una forma simplificada es un conjunto de tablas relacionadas entre sí, que conjuntamente ofrecen la información que necesita un sistema de información determinado. Asimismo, también forman parte de la base de datos otro tipo de estructuras como son índices, restricciones,...

—Tablas: Podemos considerar de una forma simplificada a las tablas como las diferentes unidades donde almacenamos la información, así pues podemos tener una tabla de documentos, otra de tipos de documentos, otra de grupos sintagmáticos... Las tablas se componen de campos, que es donde guardamos realmente la información. Así pues la tabla *TIPOS_DE_DOCUMENTOS*, podría contener dos campos: uno denominado código_de_documento y otro descripción_de_documento.

—Sistema gestor de base de datos (SBGD): Programa que facilita la realización de operaciones sobre la información contenida en una base de datos, permitiendo llevar a cabo operaciones de inserciones, borrados, consultas,... Su propósito principal es permitir al usuario almacenar, actualizar y consultar datos en términos abstractos, de forma que sea relativamente fácil mantener y obtener información de una base de datos. El *SGBD* libera al usuario de conocer exactamente la organización física de los datos y de crear algoritmos para almacenar, actualizar o consultar esa información.

—Índices: Su objetivo es el de agilizar el acceso a los datos contenidos en las tablas, así como su ordenación. Se crearán índices respecto a los campos por los cuales se considera que será muy frecuente el acceso, y de esta forma ganar en velocidad y rendimiento.

—SQL: Para proporcionar a los usuarios las diferentes facilidades, el *SGBD* aporta uno o más lenguajes especializados llamados Lenguajes de Base de Datos. Cada gestor ofrece uno diferente, aunque se puede considerar *SQL* (*Structured Query Language*) como el estándar en esta área.

—DML (*Data Manipulation Language*). Se trata de un lenguaje de manipulación de datos, orientado al proceso y extracción de información almacenada en el *SGBD*. Se usa para añadir, borrar, recuperar o modificar información de las diferentes tablas. SQL dispone de sentencias DML.

—DDL (*Data Definition Language*). Se trata de un lenguaje orientado a la descripción de la base de datos dentro del *SGBD*, así como a modificaciones de la base de datos y cambios en su estructura física. Permite la definición de las tablas (nombres, campos, dominios), índices,... así como todo lo necesario para asegurar la integridad referencial y validaciones. SQL dispone de sentencias DDL.

4.1.1. Paradox.

En primer lugar se optó por *Paradox*, que en cierta forma es el sistema gestor de base de datos más habitual para trabajar con *Delphi*. Su diseño fue bastante rápido en un inicio, ahora bien, en el momento en que se pretendía almacenar bastante información, y acceder a ella cruzando sobre diferentes tablas su rendimiento no era lo suficientemente deseable. En algunas ocasiones los ficheros llegaron a dañarse (corrupción de índices), y en algunos casos se produjo un daño total en los ficheros que contenían la información debidos a *cuelgues* en el sistema.

Es más, realmente no se trata de un sistema gestor de base de datos, se trata más bien de un sistema gestor de ficheros, un paso anterior en la evolución de los sistemas gestores de bases de datos que conocemos en la actualidad, y la forma de tener que programarla era en algunas ocasiones demasiado rudimentaria, no soportando en ciertas ocasiones comandos *SQL* (en la actualidad considerado como un estándar en el diseño de bases de datos).

4.1.2. Interbase.

Una vez descartado el uso de *Paradox*, fue necesario buscar otro sistema gestor de base de datos; los requisitos que se exigían a priori eran:

—Integración con el lenguaje de programación. Se trataba de buscar un *SGBD* que se integrara lo mejor posible con el lenguaje de programación, en nuestro caso *Delphi*, para facilitar el diseño de la aplicación y posteriores modificaciones.

—*SGBD* que dispusiera de un lenguaje *DDL (Data Definition Language)*: Básicamente que permitiera escribir en un fichero en el cual poder registrar de una forma bastante clara la definición de la base de datos, que facilitará al programador y usuarios avanzados observar la estructura de la base de datos creada y facilitar posibles modificaciones de la misma.

—*SGBD* que dispusiera de un lenguaje *DML (Data Manipulation Language)* potente: De forma simple, que tenga instrucciones que faciliten el acceso a los datos, su inserción, modificación y borrado, como es el caso de SQL.

—*SGBD* veloz en ejecución: Interesa que el *SGBD* que seleccionemos sea rápido en la inserción, modificación, borrado y búsqueda de información, ya que repercutirá directamente en la velocidad de ejecución de la aplicación.

—Requisitos de máquina: El *SGBD* a emplear debería funcionar con los mínimos requisitos de ordenador, sobre todo en lo que respecto a la memoria de disco y a la memoria *RAM*. De todas formas, dada la actual relación coste/prestaciones del mercado éste es un aspecto que no debiera revestir demasiada importancia.

—*SGBD* económico: A igualdad de prestaciones, sería preferible optar por un *SGBD* por el cual no haya que pagar costes muy elevados por cada una de las instalaciones que pretendamos realizar.

—...

El sistema *SGBD* escogido ha sido *Interbase* ya que cumple con todos los requisitos mencionados anteriormente:

—Total integración con *Delphi.*

—Potente lenguaje *DDL.*

—Potente lenguaje *DML.*

—*SGBD* gratuito (*freeware*).

—Velocidad aceptable.

—Requisitos de memoria razonables.

Otros *SGBD* como puede ser el caso de *ORACLE*, robusto y potente donde los haya, se ha descartado, por ser su coste muy elevado para el tipo de aplicación que deseamos realizar.

Ahora bien, al tratarse Interbase de un *SGBD* de tipo cliente / servidor, hay determinadas operaciones que se hacen muy lentas, como, por ejemplo, el mostrar por pantalla una lista con todas las formas de un documento y pretender ir directamente a la última forma (lo que hace es intentar leer todas las formas intermedias para acceder a la última, lo cual puede resultar mucho más lento que con un típico sistema de gestor de ficheros como era con *Paradox*).

Este tipo de problemas lo hemos solucionado recurriendo a otros componentes de acceso a *Interbase* (*IBOBJECTS*), con lo que las prestaciones en cuanto a velocidad han aumentado de forma muy considerable.

4.2. Prestaciones.

Aunque a lo largo de estas líneas, y al describir de modo general la gestación de la aplicación informática, concebida para satisfacer una necesidades concretas de un grupo determinado de investigación, se han descrito de forma implícita las prestaciones del *FRECONWIN*, nos parece interesante finalizar este artículo mediante la representación gráfica de las diferentes entidades que contempla la aplicación y como se relacionan éstas entre sí. A este modelo lo denominamos *modelo entidad-relación.*

Dada la dificultad que entraña el intentar describir en soporte papel los procesos de una aplicación informática de este estilo, de una forma más o menos breve y concisa, hemos pensado que este gráfico nos aproximará a la situación usuario/investigador → corpus → herramienta.

A continuación pasamos a explicar estos conceptos con algo más de detalle.

4.3. Modelo entidad-relación.

La aplicación informática *FRECONWIN* almacena datos referentes a documentos, tipos de documentos, formas, segmentos, grupos sintagmáticos, categorías,... A éstos elementos de información, desde un punto de vista informático, a nivel de análisis, se les denomina *entidades*. Toda esta información está inter-relacionada entre sí, y es habitual realizar el lla-

mado *modelo entidad-relación*, que se encarga de mostrar de una forma gráfica las diferentes entidades y como se relacionan entre sí. Básicamente se trata de modelar el sistema que deseamos obtener.

A partir de este modelo gráfico, es mucho más sencillo plantear cómo diseñar la base de datos, estructuras relacionadas y estudiar cómo plantear la realización de la aplicación.

El modelo entidad-relación tiene una lectura muy simple:

—Las *entidades* se muestran como *formas rectangulares* en el esquema.

—Las *entidades* se relacionan entre sí. Gráficamente esto se representa por un *rombo.*

—Los números que hay a ambos lados de las relaciones indican la *cardinalidad.* Habitualmente se trata de una pareja de valores mínimo y máximo. Así pues la relación entre *Tipo_de_documento* y *Documento* se lee en un sentido de la siguiente forma: De un *Tipo_de_documento* puede haber 0 ó N documentos, y un *Documento* es de 1 tipo determinado (obligatoriamente), leyéndolo en el sentido contrario.

Veamos un sencillo ejemplo:

La entidad *TIPOS_DE_DOCUMENTOS* podrá almacenar información del siguiente estilo:

—Discursos políticos

—Obras de teatro del siglo XIX

—Obras de teatro del siglo XVIII

—...

La entidad *DOCUMENTOS* podrá almacenar información del siguiente estilo:

—Discurso del político x1 de fecha y

—Discurso del político x1 de fecha z

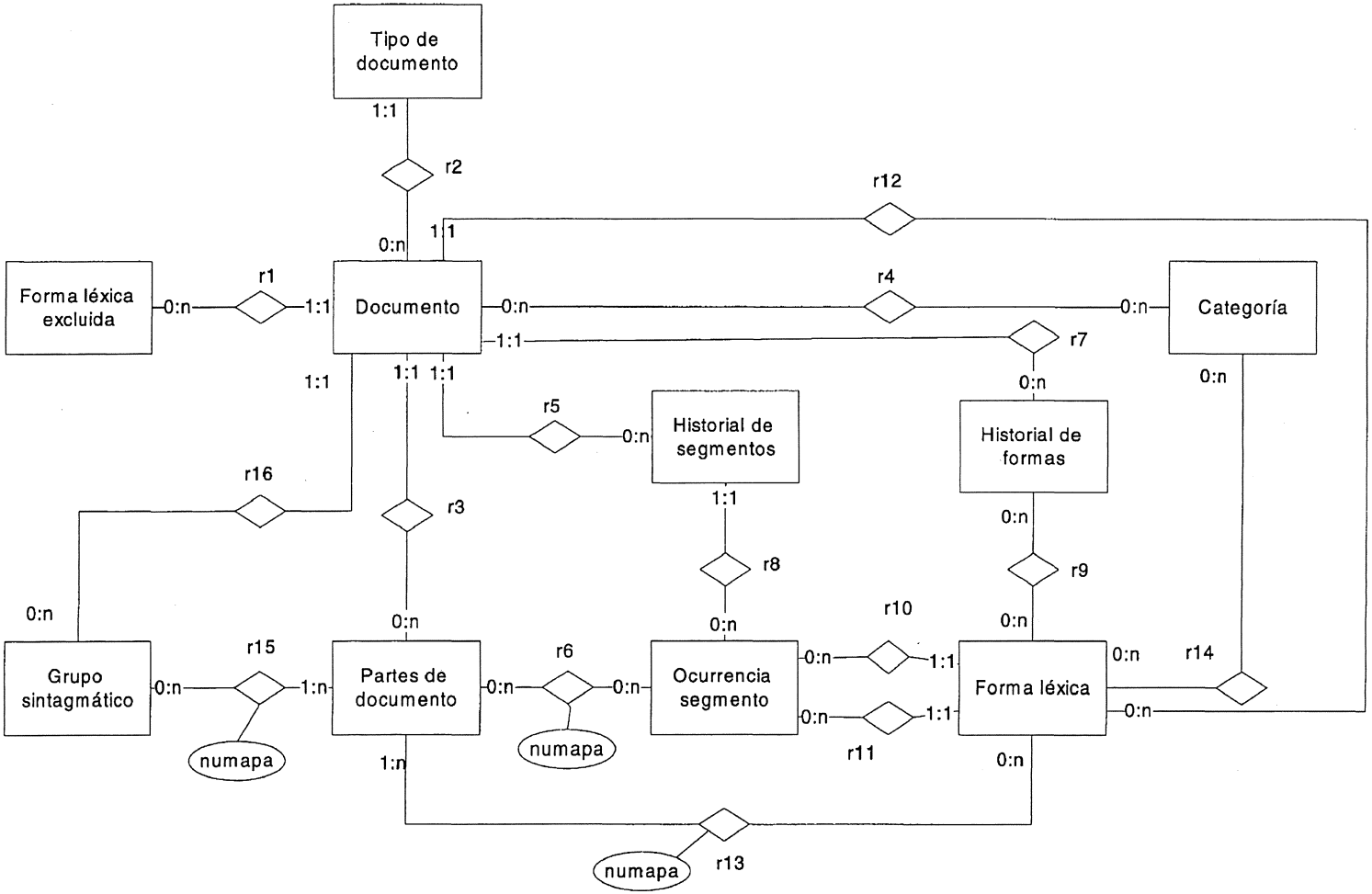
—Discurso del político x2 de fecha w

—Obra xx

—Obra yy

—...

Así pues, está claro, que 'Discurso del político x1 de fecha y' estaría relacionada con 'Discursos políticos', de ahí la relación entre *DOCUMENTOS* y *TIPOS_DE_DOCUMENTOS*, y que indica que un *DOCUMENTO* es de un *TIPO_DE_DOCUMENTO*, pero que de un determinado *TIPO_DE_DOCUMENTO* puede haber muchos documentos. En nuestro caso del *TIPO_DE_DOCUMENTO* 'Discursos políticos' puede haber muchos *'DOCUMENTOS'*.


Tipo de documento
1:1
r2
0:n
r12
1:1
Forma léxica excluida
0:n
r1
1:1
Documento
0:n
r4
0:n
Categoría
1:1
r7
0:n
1:1
1:1
1:1
r5
0:n
Historial de segmentos
Historial de formas
0:n
r16
r3
1:1
0:n
r8
0:n
r9
0:n
0:n
0:n
r10
r15
r6
0:n
Grupo sintagmático
0:n
1:n
Partes de documento
0:n
0:n
Ocurrencia segmento
0:n
1:1
0:n
1:1
Forma léxica
0:n
r14
0:n
numapa
numapa
r11
1:n
0:n
numapa
r13

A continuación, pasamos a describir brevemente cada una de las entidades del *modelo entidad-relación*, la mayoría de ellas bastante claras por el nombre de las mismas:

Entidad	Descripción
Documento	Relación de documentos que serán tratados por la aplicación
Partes_de_documento	Contiene información acerca de las diferentes partes en que dividimos un documento
Categoría	Relación de las diferentes categorías en que pueden agruparse las diferentes formas léxicas
Tipos_de_documento	Relación de los diferentes tipos de documentos, que podrán ser asociados a los documentos
Forma léxica	Relación de las diferentes formas léxicas contenidas en los distintos documentos a tratar por la aplicación
Forma léxica excluida	Relación de formas que excluimos para cada uno de los documentos
Historial de segmentos	Permite almacenar las diferentes búsquedas de segmentos dentro del documento que tienen interés para el tratamiento del mismo
Historial de formas	Permite almacenar las diferentes búsquedas de formas dentro del documento que tienen interés para el tratamiento del mismo
Ocurrencia segmento	Mantiene información sobre los segmentos (combinación de forma segmento y forma polo para una determinada búsqueda o historial de segmentos)
Grupos sintagmáticos	Combinación de formas léxicas consecutivas de un documento.

Cada una de estas entidades contiene información, y ésta se encuentra estructurada en elementos individuales de información denominados *atributos* (concepto bastante similar al de *campos*, cuando más arriba hacíamos una breve explicación del concepto de *tablas*). A continuación observamos cuáles podrían ser los atributos de algunas de estas entidades:

—**Documento:**

—código,

—nombre,

—observaciones,

—path_documento,

—alineacion,

—diferencia_may_min,

—excluir_numeros,

—bloqueado,

—...

—**Tipo_de_documento**

—codigo

—descripción

—...

—**Categorías:**

—código,

—descripción

—...

Las entidades se relacionan entre sí a través de las relaciones. Algunas de ellas también almacenan información, como puede ser el caso de *r6* que indica el número de apariciones de una determinada ocurrencia de un segmento, en una determinada parte de un documento. En estos casos, le damos un nombre más significativo a dicha relación.A continuación mostramos las diferentes relaciones que aparecen en el *modelo entidad-relación.*

Relación	Descripción
r1	Relaciona las formas léxicas excluidas de un determinado documento, para intereses específicos del tratamiento de dicho documento.
r2	Relaciona de qué tipo es un determinado documento.
r3	Permite detectar en qué partes se divide un determinado documento.
r4 = Categorías_de_documento	Relaciona las diferentes categorías a que pueden asociarse las formas léxicas de un documento en concreto.
r5	Relaciona las diferentes búsquedas de segmentos con los documentos.
r6 = Segmentos_partes	Indica información relativa a la aparición de los diferentes segmentos de una búsqueda (historial) respecto a las diferentes partes del documento, y si lo creemos oportuno información sobre valores estadísticos.
r7	Relaciona las diferentes búsquedas de formas con los documentos.
r8	Relaciona que segmentos se corresponden a una determinado criterio de búsqueda de segmentos.
r9 = Ocurrencia_Forma	Formas léxicas asociadas a una determinada búsqueda de formas/expresiones (histórico de formas), y si lo creemos oportuno información sobre valores / resultados estadísticos.
r10 = Forma_polo	Relaciona cual es la forma polo de una determinada aparición de un segmento, que cumple con los criterios de una determinada búsqueda (historial) de segmentos.
r11 = Forma_segmento	Relaciona cual es la forma segmento de una determinada aparición de un segmento, que cumple con los criterios de una determinada búsqueda (historial) de segmentos.
r12	Relaciona cuáles son las diferentes formas léxicas de un documento.
r13 = Formas_partes	Indica información relativa a la aparición de las diferentes formas léxicas respecto a las diferentes partes del documento
r14 = Categoría_forma	Relación que permite asociar las diferentes formas léxicas a las categorías
r15 = GrupoSintagmatico_partes	Relación que nos indica el número de apariciones de los diferentes grupos sintagmáticos respecto a las diferentes partes del documento.
r16	Relaciona los grupos sintagmáticos con el documento al que corresponden.

Un proceso de paso del *modelo entidad relación* al *modelo relacional* y seguidamente una normalización de las relaciones resultantes nos ofrecerá la estructura de las tablas a crear para el desarrollo de la aplicación.

BIBLIOGRAFÍA

BERNARD MICHEL (1999): *Introduction aux études assitées par ordinateur*, Paris: PUF.

BRUNET, E. (1990): "Apport des technologies modernes à l'histoire littéraire", en Béhart y Fayolle, *L'Histoire littéraire aujourd'hui*. París, Armand Colin

HYPERBASE (Version 2.0 para Windows, Abril 1998), Étienne Brunet.

KNUTH (1981): *The Art of Computer Programming, Seminumerical Algorithms*. Vol 2. Second Edition. Addison-Wesley Publishing Company

LE SPHINX (Version 2000) – le Sphinx Développement. Seynod, France.

MARCO CANTÙ (2000): *La biblia de Delphi 5*, Anaya Multimedia

SPAD-T: *Système portable pour l'analyse de données textuelles*. París: CISIA ed.

TACT (*Text Analysis Computing Tools*, versión 2.1 – 1993) – Centre for Computing in the Humanities, Faculty of Arts and Science (Universidad de Toronto). MICHAEL STARS, JOHN BRADLEY, EDWARD HEINEMANN, JOHN C. HURD

VÍCTOR MORAL (1999): *Delphi 4*, Prentice Hall